DB 設計を支援する情報資源辞書システムの操作機能と実現法†

関根 純†† 川下 満†† 中川 優††

データ中心アプローチの普及と共に，DB の全社的な体系化の重要性が認識されてきた．これを達成するためには，DB 設計支援システムと DB 設計情報を集中管理する辞書（ディクショナリ）システムが不可欠である．そこで，本論文では，DB 設計支援システムからの要求に基づき，ISO が標準化を進める情報資源辞書システム (IRDS) の必要機能を具体化した．その結果，プロジェクト管理のためのワーキングセット管理機能について新たな機能提案を行うと同時に，その機能のシンタクスを SQL の拡張として提案した．また，従来の工程管理機能のみでは実際の工程をうまく定義できないこと，設計情報の品質まで管理することはできないことを示し，工程管理機能の拡張と新たに品質管理機能を提案した．さらに，DB 設計作業の特質から，IRDS では参照一貫性の波及範囲を限定すべきことを明らかにし，その対処法を示した．最後に，IRDS におけるディクショナリの設計で工夫すべき点，ならびに，IRDS の一実現方法を示した．以上の提案に基づき，IRDS の一部機能とその上で動作する設計支援システムを実現し，運用，評価により，本 IRDS が実用に十分耐えることを確認した．

1. はじめに

計算機システムの実現に向けて体系的なシステム設計支援と設計情報の集中管理のためのディクショナリシステム[9),24)]が重要であるとの認識が広まっている．ディクショナリシステムとは，設計情報を管理するために，設計情報の版を管理する機能，設計情報をプロジェクトごとに管理する機能，設計作業の工程を管理する機能，あるいは，設計情報の構成を管理する機能などをデータベース管理システムに強化したものであり，リポジトリマネジャ[27)]のような製品もいくつか出現している．一方，ISO[15),16)]，ANSI[1),2),11)] では情報資源辞書システム（Information Resource Dictionary System, IRDS と略す）の名称で標準化を進めてきている．

現在，筆者らは DB の全社的な体系化を支援する DB 設計支援システムを開発中であり，特に DB 設計情報を集中管理するディクショナリシステムの実現は，そのための重要な課題の1つとなっている．そこで DB 設計支援分野での要求を満たすディクショナリシステムとはどのようなものであるかを分析し，提案されている技術と比較評価を行った．

今回，筆者らは，ISO の IRDS が提案する概念を基本に，DB 設計支援分野での要求を満たす IRDS の操作機能を開発した．その評価によると，プロジェクト単位の設計情報管理機能，設計情報の工程管理機能が不足していること，および，一貫性制約の考え方が DBMS とは異なり，一貫性制約の保証範囲を限定すべきことがわかり，ISO の提案に対するこれらの拡張を行った．

2章では，DB 設計支援分野の概要と要求，IRDS 関連研究の動向，および，本論文でのアプローチを述べる．3章では，提案機能と DB 設計支援システムへの適用例を示し，4章では，ディクショナリ設計上の考慮点について述べる．最後に5章では，IRDS の実現法に対する基本的な考え方を述べる．

2. 研究への要求とアプローチ

2.1 DB 設計支援の概要

本節では，対象とする DB 設計支援システムの機能と設計情報の概要を示す（**図 1**）．DB 設計支援システムは，大きく，4つのツールから構成される．DB 概念設計ツールは，テーブル，データ項目，関数従属性などの DB 情報を用いて，データの正規化を行う．DB 論理物理設計ツールは，テーブル，データ項目などの DB 情報に加え，どのアプリケーションがどのテーブル，データ項目をアクセスするかの DB アクセス情報を用いて，テーブルとインデックスの性能チューニングを行う．データ標準化ツール[29)]は，社内の全データ項目の標準化を図るため，システムにまたがり，データ項目の名前，属性，桁数，コードをチェックし，それらの統一を図る．このツールは，DB 情報に加え，社内で共用される統一コード，および，データ項目名として使用が許される標準の用語に関する情

† Operations and Implementation of Information Resource Dictionary System for DB Design by JUN SEKINE, MITSURU KAWASHIMO and MASARU NAKAGAWA (NTT Network Information Systems Laboratories).

†† NTT 情報通信網研究所

報を取り扱う．DB 定義文生成ツールは，DB 情報を用いて特定 DBMS 向けに DB 定義文を出力する．

4つのツールは，基本的には次の手順で使用される（図 2）．①まず，DB 設計者は DB 概念設計ツールを用いて DB の構造の概要を設計し，②次に，アプリケーション設計者は DB 論理物理設計ツールを用いて DB アクセス情報を追加する．③DB 設計者は，この DB アクセス情報を元に，同ツールを用いて DB の構造の性能チューニングを行う．この後，作成された DB 構造に対して，④プロジェクト内のデータ管理者はデータ項目名が標準の用語の正しい組み合わせで成っているかを，一方，⑤管理部門のデータ管理者は，全社的に統一されたデータ項目名が使用されているか，あるいは，データ項目が統一コードを使用しているかなどをチェックする[8),30),31)]．プロジェクトで新たに使いたい標準用語は，データ管理者に申請し，受諾されて初めて共通の標準用語に登録される．統一コードも同様にデータ管理者によって審査される．⑥審査終了後，DB 定義文生成ツールにより，DB の定義文が出力される．

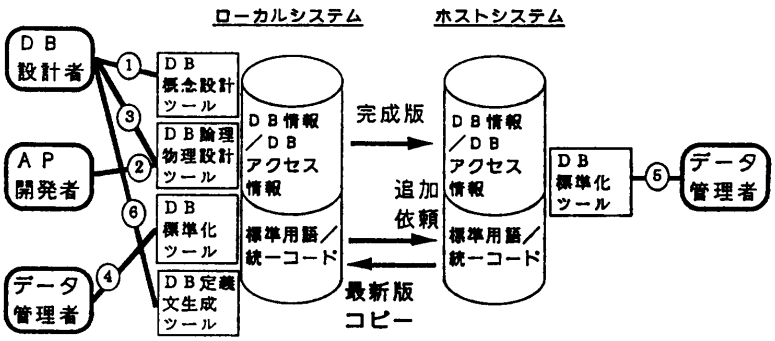


図 2 DB 設計手順の概要
Fig. 2 General flow of database design.

本システムの構成は，プロジェクトと管理部門が地域的に離れていることを前提に，標準用語と統一コードを管理するホストシステムと作業中の DB 情報を管理するローカルシステムからなる分散形態で実現した．ローカルシステムにおいて抽出される新たな標準用語，および，統一コードの候補は，ローカルシステムのディクショナリに格納され，その後，ホストに送付されて初めて審査を受ける．審査が完了した後に，共通のディクショナリに登録され，新しい標準用語，および，統一コードの版としてローカルシステムにコピーが再送付される．DB 情報は，プロジェクトでの設計が完了した後，ローカルシステムからホストに転送され，ホストで管理される．

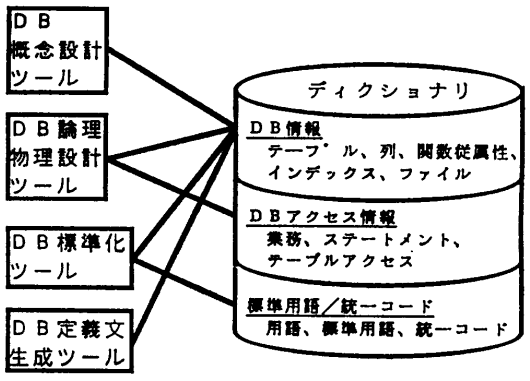


図 1 DB 設計支援システムの概要
Fig. 1 Architecture of the database design system.

以上のツール群を支援する IRDS への要求，関連研究の現状，および，アプローチなどを次に示す．

2.2 プロジェクト単位の設計情報管理機能

前節で述べた手順による DB 設計は，通常，プロジェクト単位で行う．どのような設計単位をプロジェクトに割り当てるかは状況により変わり，DB（およびそれに含まれる全テーブル），DB 内の複数テーブルなど，多様である．同一のプロジェクトでも，時間と共に，さらにサブプロジェクトに分解されることがある．したがって，IRDS は，プロジェクトごとに設計情報をグループ化し，時間と共にプロジェクトの構成が変わっても，追随してグループの構成を柔軟に変更できる必要がある．また，設計はプロジェクトに閉じて行うため，プロジェクトの範囲に限定した操作機能も必要である．

設計情報のグループ化については，設計情報をオブジェクトにより表現し，さらにそのオブジェクトを「複合オブジェクト」にグループ化して操作を行う提案がある[4),12),19),22)]．この提案によれば，例えば，複合オブジェクトに DB を対応づけることになる．しかし，この提案では次の2つの問題がある．

①複合オブジェクトは，あらかじめ構造がはっきりしている設計情報の表現のための概念であり，含まれるオブジェクトのタイプは事前定義が必要である．ところが，プロジェクトが管理する設計情報は，DB，

テーブルなど，時間と共に内容が変化するため，事前の定義ができない．

②プロジェクトの範囲に限定して操作を行う機能を具体化した文献に4),19)がある．これらの文献では，複合オブジェクトに含まれるオブジェクト群の参照情報（reference）を，複合オブジェクトの繰り返しを許す属性として管理する手法を用いているが，この手法では，複合オブジェクトに限定して要素のオブジェクトを操作する際，複合オブジェクトを識別した後，その中の参照情報を経由して要素オブジェクトをアクセスするという，アクセスパスを考慮した操作の記述が必要となり，複雑である．さらに，要素となるオブジェクト間の結合処理の場合，これをオブジェクトの種類ごとに行う必要があるため，より複雑となる．

一方，ISO の IRDS では，複数のオブジェクトをグループ化するワーキングセット（WS と略す）の概念が提案されている．WS は，任意個のオブジェクトを格納する入れ物として定義され，オブジェクトの参照，更新操作を行う前に，まず，オブジェクトが存在する WS の名前を指定する必要がある．このような WS は1つのディクショナリに複数個作成できる．

しかし，WS については，設計情報を構造化したものと位置づけるのか，プロジェクトの管理単位と位置づけるのかが明確でなく，したがって，機能提供の方針も明確でない．

そこで本論文では，①の問題を解決するため，設計対象である複合オブジェクトと，プロジェクト管理のためのオブジェクトのグループ化の概念を分けるという方針に立ち，この WS により後者のプロジェクト管理を行うことにする（図3，詳細は3.1節に示す）．すなわち，複合オブジェクトと WS は別個の概念と位置づける．これに伴い，参照，更新などの従来の DBMS にある機能の拡張，新たに必要となる WS 単位の操作機能，および，セキュリティ機能を提案する．操作機能は②を改善するためグループ化操作と，範囲を限定した操作を容易に記述できる言語仕様とする．操作機能の記述は，ISO の IRDS がリレーショナルモデルを基本としていることから，SQL に対する拡張として表現する．以後も，IRDS のデータモデルについては，SQL の用語を使う．

2.3 設計情報の工程管理機能

大規模な設計は段階的に進むものであり，設計工程の進捗，設計情報の品質の程度を設計情報と関係づけて管理する必要がある．DB 設計にも，例えば，概念設計，論理設計，および，物理設計などの工程がある．

前述の ISO の IRDS[15] では，WS 単位にどの工程に属するかを設定できる．この工程定義には，設計途中を表すアンコントロール，完成状態を表すコントロール，および，一度完成した古い設計情報の版が存在するアーカイブがある．アンコントロールはさらにユーザが細分化することができる．WS の工程は，ある工程から，あらかじめ定めた別の工程にのみ変更することができる．これを工程間の遷移と呼ぶことにする．工程をノード，工程間の遷移をノード間の弧とするグラフ（これを工程遷移グラフと呼ぶことにする）を考えると，ISO の IRDS では，工程遷移グラフは，木の根をアーカイブ，次にコントロール，さらに，アンコントロールに属する複数の工程が接続する，木構造に制約される（図5(a)）．

しかし，この制約に従うと次の3つの問題が生じる．①工程遷移グラフは木構造とは限らない．例えば，新規システムの DB 設計と，既存システムの更改に伴う DB 設計の工程間の遷移は異なり，両者を考慮した場合，工程遷移グラフは，ネットワーク構造である必要がある（図4）．②設計の工程は設計情報ごとに異なるため，工程と工程間の遷移も設計情報の種類ごとに分ける必要があるが，現状では設計情報ごとに工程を定義できない．例えば，DB 情報の設計工程とプログラム情報の設計工程は異なる．③工程は，大きな設計フェーズの設定には使用できるが，設計情報の完成度を示す品質を表すには，さらに次の2つの問題が生じる．1つの問題は，設計情報の品質を設定する単位は WS より小さく，しかも，全体の工程で期待される設計品質よりも部分的に進んだ品質にある設計情報が存在することである．例えば，DB 設計ではクリティカルなポイントについて実現性の見通しを得るため，部分的に先の工程の設計作業を行うことがあるため，このような状況が生じる．次の問題は，品質の尺度が複数存在する場合，1つの工程ではそれを表しきれないことである．例えば，DB 設計では，各々のデータ項目がどの程度標準に従った名前を付与されているかを表すデータ標準化の観点からの品質と，各々のテーブルに対する性能設計の観点からの品質があり，それらは互いに独立である．

このような ISO の IRDS における3つの問題に対する解決法を以下に示す．

①については，工程遷移グラフをネットワーク構造

とすることで対応できる．これは文献6)と同じ方針であるが，6)では記述されていないディクショナリとの関連を明らかにする．②については，設計情報の種類ごとに独立な工程遷移グラフを定義可能とすることで対処する．③については，工程と，設計品質が1つの概念で表せないことは明らかであるので，後者について品質管理という新たな機能を提案する．

なお，設計情報の品質を制御するため，複合オブジェクトの版の状態を作業状態 (transient)，施行状態 (working, released) に分ける方式[5),17),20)]があるが，表現できる工程の遷移状態が単純で工程管理に使用できないため，拡張する．

2.4 一貫性制約制御機能

DBMS の一貫性制約については，既に多くの論文[3),7),14),23),24),26),28)]で言及され，アサーション，トリガ，参照一貫性などが提案されている．しかし，このような一貫性制約の機能をそのまま IRDS に持ち込むと次の2つの問題が生じる．①設計作業は様々なツールを介して行われるが，ユーザに見せる設計情報の範囲は通常限定されている．ところが，参照一貫性に代表されるような，変更の影響が複数のテーブルに及ぶ一貫性制約は，ユーザに見せる範囲を越えて影響が波及する可能性がある．すなわちあるツールの使用により設計者に見える範囲外の設計情報に影響が及ぶのは，ツールの動作に対する設計者の理解を困難にするため，設計作業においては一貫性制約を損なう以上に問題である．②通常，ディクショナリには，管理責任者の異なる設計情報が混在している．しかも，それらの設計情報の間に関係がある場合がある．例えば，統一コードはデータ管理者が管理し，それをデータベース設計者が管理する複数の DB で共用する場合である．この時，統一コードを削除するとデータベース設計者の DB 情報に影響が出る．このように，ツールを使用している設計者のアクセスの権限外に一貫性制約の影響が波及するのは，セキュリティの点から問題である．

本論文では，一貫性制約を保証する範囲を，ツールが設計情報を参照する範囲とユーザの権限が及ぶ範囲に制限することにより，これらの問題を解決する方法を提案する．

3. IRDS の機能概要

3.1 ワーキングセット管理機能

(1) ワーキングセットの定義，生成機能

リレーショナルモデルを基本とした IRDS では，設計情報は複数テーブルの複数タプルにより表現される．WS は，これらの複数のテーブルのタプルを格納する入れ物として定義する．柔軟な WS を提供するためには，WS に任意の種類のテーブルのタプルを制限なく格納できることが望まれるが，他方，WS に格納される設計情報の種別を DB 情報，コード情報のように分類し制限できると，DB 設計支援システムにとっては，その分類に応じた処理と工程管理ができるため利便性が高い．ここでは，後者のメリットを重視して，WS を WS タイプに分け，WS タイプごとに格納可能なテーブル名を WS 定義時に指定可能とした．

WS，および，WS タイプに対する操作をコマンド言語風に記述し表1の項番1から5に示す．WS タイプの定義は define working set type により，WS の生成は create working set により行う．WS の削除は，操作を誤ると影響が大きいため，確認の意味で仮削除 remove working set と本削除 delete working set の2段階の操作を経て削除される．仮削除は，いつでも仮削除の解除 restore working set により取り消せる．

(2) セキュリティ機能

WS はプロジェクトの資産であるため，WS のアクセスを特定のユーザにのみ許す制御機能は必須である．ここでは，WS 単位で，アクセス権限を付与可能とした．権限の種類とその効果は次のように整理される．

IRDS 管理者のみが WS タイプを定義でき，IRDS のユーザを登録できる．IRDS 管理者は，ユーザに，特定 WS タイプに属する WS の作成権限を permit creation working set type により与える（表1項番6，7参照）．WS を作成したユーザは，その WS の更新権限をもつと同時に，permit access working set により，他のユーザに更新権限，あるいは，参照権限を与えることができる．WS の更新権限（参照権限）をもつユーザは，WS に含まれるテーブルのタプルの参照，更新（参照）が可能である．

なお，WS 更新権限を持っているユーザにのみ 3.2 節で述べる WS の工程の遷移が可能である．

(3) ワーキングセットの使用宣言機能

ある WS に含まれるテーブルのタプルの操作は，open working set により参照，あるいは更新モードでの WS の使用を宣言した後に可能となる（表1項番8，9参照）．この時，権限がチェックされる．WS

の使用宣言と共に，当該 WS には，参照，または，更新のロックが掛けられる．このロックは，トランザクションにまたがる長期ロックである[21]．文献 21) に示すロック種別間の関係からの拡張点は，ユーザが複数のプログラムで，同じ WS に対する操作を可能とするため，同一ユーザに限り，同じ WS に対する参照と更新を受け付けることである．

IRDS の直接のユーザである設計支援システムは，同時に複数の WS を使用宣言可能である．これにより，標準用語が含まれた WS を参照しながら，DB 情報が含まれた別の WS で更新処理が可能になる（図3，詳細は(7)に示す）．これは，カレントな WS 1つしか同時に操作できない ISO の IRDS に比し，大きな利便性を提供している．使用が終了したら close working set で使用終了宣言を行う．

（4） ワーキングセットの参照，更新機能

ある WS に限定したテーブルのタプルの参照，更新を実現するために，ここでは，従来からある SQL の参照，更新機能を次のように変更した．参照では，WS を限定する句が追加される．

```
select * from data_item
where item_name = "回線名"
within 〈WS 名〉 (〈版番号〉), …
```

within 句により，検索する範囲の WS を限定する．なお，WS は，版管理を可能とするため WS 名と版番号で一意に識別可能である．結合処理，副照会を行う場合，複合オブジェクトに関する提案[4),19)]ではテーブルごとに複合オブジェクトを指定する操作が必要となるのに対し，一回の within 句指定で済むため，操作が簡易化される．さらに，複数の WS を同時に宣言することができるため，DB 概念設計，論理物理設計ツールでは，1つの DB に対応する1つの WS に限定した操作が行える一方，データ標準化ツールでは，複数の DB にまたがって同一の名前のデータ項目が存在するかをチェックするため，複数の WS を横断的に検索することが可能となる．なお，参照，更新の時点において，WS は使用宣言をされている必要がある．

追加，変更，削除操作においても，WS を限定する必要があるため，SQL 文に within 句を付加する．ただし，追加の場合，挿入対象の WS は，1つであるため，指定可能な WS は1つである．なお，WS の導入により，ユニークインデックスにより保証されるユニーク性の範囲は，WS 内に変更される．

```
insert into 〈テーブル名〉 values (〈値リスト〉)
    within 〈WS 名〉 (〈版番号〉)
```

（5） ワーキングセット一括操作機能

以上の基本コマンドに加え，WS に含まれる設計情報の管理単位を柔軟に変更する次の WS 一括操作が可能である（表1項番 10 から 14 参照）．

①WS 間での内容のコピー (copy working set)

②WS の内容の他 WS への移動 (move working set)

③WS の内容の全削除 (clear working set)

④WS のロード／アンロード (load/unload working set)

表 1 主な IRDS 操作機能
Table 1 IRDS operations.

#	操作機能名	操作機能シンタックス
1	WSタイプ定義	define working set type <WSタイプ名> containing {<テーブル名1>、}
2	WS生成	create working set <WS名> (<版番号>) in <WSタイプ名> phase <工程名>
3	WS仮削除	remove working set <WS名> (<版番号>)
4	WS本削除	delete working set <WS名> (<版番号>) [with check option]
5	WS仮削除取消	restore working set <WS名> (<版番号>)
6	WS作成権付与	permit creation working set type <WSタイプ名> user <ユーザ名>
7	WSアクセス権付与	permit <参照／更新モード> access working set <WS名> (<版番号>) user <ユーザ名>
8	WS使用宣言	open working set <WS名> (<版番号>) with mode <参照／更新モード>
9	WS使用終了宣言	close working set <WS名> (<版番号>)
10	WSコピー	copy working set <WS名1> (<版番号1>) to <WS名2> (<版番号2>)
11	WS移動	move working set <WS名1> (<版番号1>) to <WS名2> (<版番号2>)
12	WS内容消去	clear working set <WS名> (<版番号>)
13	WSロード	load working set <WS名> (<版番号>) from file <ファイル名>
14	WSアンロード	unload working set <WS名> (<版番号>) to file <ファイル名>
15	ビュー定義	define view <ビュー名> {working set <WS名> (<版番号>) {phase {<工程名>、} with mode <参照／更新モード>、}、}
16	ビュー使用宣言	open view <ビュー名> with mode <参照／更新モード>
17	ビュー使用終了宣言	close view <ビュー名>
18	工程定義	define phase <工程名1> in lcp <フェーズ> working set type <WSタイプ名> after phase {<工程名2>、} before phase {<工程名3>、}
19	WS工程遷移	move lcp working set <WS名> (<版番号>) from phase <工程名1> to phase <工程名2>
20	品質セット定義	define quality indicator set <品質セット名> value {<品質名1>、}
21	品質設定	modify quality indicator set value <品質名> table <テーブル名> where <条件式> within <WS名> (<版番号>)

(注) {}内は、繰り返し可能な項目であることを表す。

(6) ビューの定義機能

DB 設計支援のツールには，複数の WS が揃って初めて機能を実行できるものが多く，これらをまとめてグループ化すれば利便性が高い．特に，個々の WS に対するオープンのモードと工程に関する実行可能条件のチェックが極めて複雑であるため，これをすべて IRDS に委せることがツール構築上効果的である．そこで，WS の任意個の集合をビューと呼び新たに定義する．ビューに対して使用可能な操作機能は，ビューの定義 define view，使用宣言 open view と使用終了宣言 close view である（表1項番15から17参照）．ビューの定義では，要素となる WS とその参照／更新モード，工程名の組み合わせを列挙する．ビューの使用宣言は，要素となるすべての WS が指定の工程にあり，指定の参照／更新モードで使用宣言可能な時のみ可能である．ビューの定義に違反する WS の工程変更，アクセスモードは制限される．また，ある時点で使用可能なビューはただ1つである．ただし，ビューを使用しなければ WS をアクセスできないという制約は設けていない．

(7) ワーキングセット定義例

DB 設計支援システムでは，次の WS タイプを定義する．図3に概要を示す．

DB 情報，DB アクセス情報は一括して WS タイプ「DB」とした．共用の標準用語，および，プロジェクト内でのみ使用し，共用の標準用語への追加の候補となるローカルな標準用語は，移行の単位でもあるため，WS タイプ「用語辞書」として「DB」と独立に実現した．同様に，共用とローカルの統一コードは，独立な WS タイプ「統一コード」として実現した．「DB」に属する WS と，ローカルの標準用語を含む WS，および，ローカルの統一コードを含む WS は1対1に対応している．

以上の WS を基本に，データ標準化ツールでは，共用，ローカルの標準用語を格納するそれぞれの WS，DB 情報を格納する WS，共用，ローカルの統一コードを格納するそれぞれの WS を対にしてビューを定義した．また，DB 論理物理設計ツールでは，同一システム内の DB 情報を格納する WS と共用／ローカルの統一コードを格納するそれぞれの WS を対にしてビューを定義した．

3.2 工程管理機能

(1) 工程遷移の定義機能

WS 単位の工程間の遷移には，2.3 節に述べたように，2つの問題がある．ここでは，その状況を例により示す．

まず，工程遷移グラフは木構造では表せないという問題がある．これを，WS タイプ「DB」に属する WS の工程定義例により示す（図4）．DB の新規設計の場合，既存システムの更改の場合，および，単に既存の DB の設計情報を登録する場合で，開始する工程は同じだが，工程遷移が異なることがわかる．なお，一括 DB 登録と会話 DB 登録の違いは，前者が設計情報のチェックを投入後バッチで行うのに対し，後者は投入時に行うことである．

さらに，工程遷移グラフが，連結した1つのグラフ構造ではなく，連結でない複数のグラフ構造となることを，WS タイプ「用語辞書」に属する WS の工程定義例により示す（図5）．用語辞書はローカルと共用の

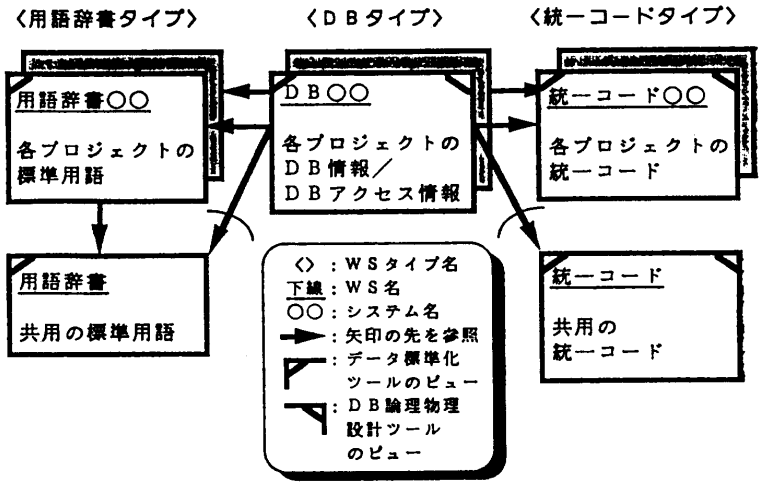


図3 WS の実現例
Fig. 3 Definitions of WS's.

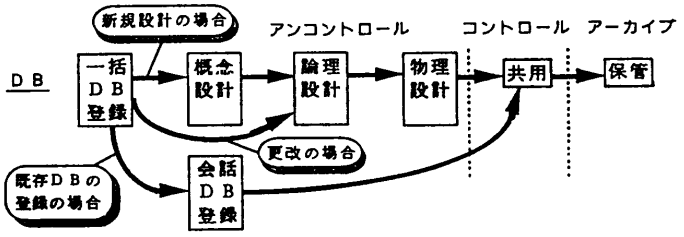
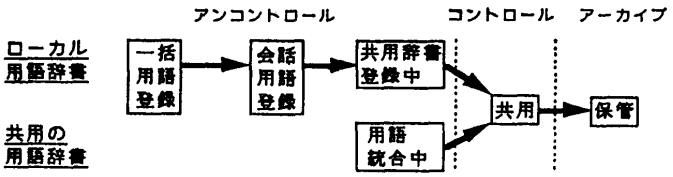


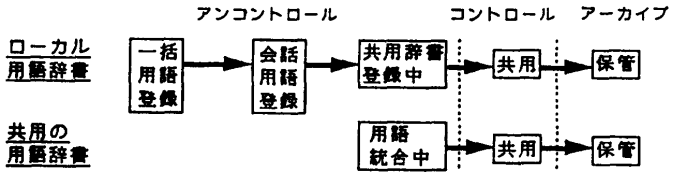
図4 工程の定義例（1）
Fig. 4 Definitions of life cycle phases (1).

用語辞書に分かれ，それぞれ別の工程遷移を持つ（図5（a））．ローカルの用語辞書は，一括／会話用語登録，共用辞書登録中，共用という遷移状態を，一方，共用の用語辞書は，用語統合中，共用という工程遷移を持つ．ここで，ローカルの用語辞書の構築が終わり，一度共用の工程に遷移したとする．従来の工程定義はコントロールに属する工程が1つのため，用語辞書修正のため，工程を戻すと関係のない工程（例えば，共用の用語辞書のための工程，用語統合中）が選択可能となり管理が不十分になる．これを防ぐため，図5（b）に示すように，コントロールを分け不要な工程への遷移を不可とした．



（a） 現 IRDS における工程定義例

（a） Definitions of life cycle phases in current IRDS.



（b） 工程定義の改善の例

（b） Improvement in definitions of life cycle phases.

図5 工程の定義例（2）

Fig. 5 Definitions of life cycle phases (2).

またもう1つの問題に，設計情報の種類ごとに，工程と工程遷移が異なるというものがある．これについては，上記の図4，図5の例で分かるように，WS タイプ「DB」の WS と，WS タイプ「用語辞書」の WS の工程と工程遷移が異なることで説明できる．

以上2つの問題を考慮して，工程は，WS タイプごとに工程遷移をネットワーク構造で定義可能とするとともに，同一の WS タイプでも連結でない複数のネットワーク構造に属する工程を定義可能とした（表1項番 18，19 参照）．この定義により，異なる WS タイプの工程には移行不可にできる．

図1に示す DB 設計支援ツールから，図4の工程は次のように使用される．概念設計，会話 DB 登録とそれより前にある一括 DB 登録にある WS のみが DB 概念設計ツールを，物理設計とそれより前の工程にある WS のみが DB 論理物理設計ツールを使用可能である．一方，DB 標準化ツールは，アンコントロールの任意の工程で使用してよい．このようにツールは，本来実施すべき工程に加え，それより前の工程でも動作可能である．これは，前の工程において，見通しを得るため部分的に後の工程の設計を先行して実施するためである．

（2） 品質管理機能

設計情報の品質の設定は 2.3 節に述べたように WS より小さい単位である必要があるため，例えば，任意のディクショナリのテーブルのタプルにあらかじめ定義した品質を設定できる品質管理機能を導入した．これは，工程とは独立に設定できる．品質情報は品質の名前で表す．テーブルごとに付与可能な品質名の集合を定義でき，品質セットと呼ぶ．品質セットは1つの評価尺度に対応し，1つのテーブルに複数指定可能である．同一のテーブルに異なる品質セットに属する品質名を複数与えることができる．品質セットの定義，品質名をテーブルに付与する機能の概要を表1の項番 20，21 に示す．

以上の機能を用いて，次のような制御を行える．例えば，品質セット名として「DB 標準化」を選び，その中に「未標準化」，「標準用語抽出完了」，「データ項目名標準化完了」を設定すると，その値に応じて DB 標準化ツールの動作を制御できる．すなわち，DB 設計の工程と無関係に，どの程度データ項目の名前の標準化が進んだかをテーブルごとに品質管理できる．

3.3 一貫性制約制御機能

IRDS では，2.4 節に述べたように，一貫性制約の影響が及ぶ範囲をツールの参照範囲かつユーザの権限の範囲に限定する必要がある．このような限定が必要な理由は2つある．①1つの理由は，ツールの守備範囲外の設計情報に影響する一貫性制約の保証まで IRDS では行えないことである．これは，従来の DBMS では一貫性制約の保証は DBMS により行うことができたが，設計支援ツールで必要とする一貫性制約の保証はより複雑であり，設計者がツールを用いて追加の情報を IRDS に与えない限り保証できないことによる．②またもう1つの理由は，設計者は，設計作業を分担しており，自らの分担と権限の範囲外の設計変更まで実施可能とは限らないことである．

例えば，図3において，「DB」は DB 設計者が管理する．また，「統一コード」に含まれた統一コードは，データ管理者が管理する．ここで，データ管理者がデータ標準化ツールを用いて，ある統一コードを削除したとする．一貫性を保つためには，この統一コードを使用している DB 情報に，別の統一コードが付与される必要がある．しかし，コードの付与は，DB 論理物理設計ツールを用いた DB 設計者の仕事であり，統一コードを削除したデータ管理者にはできない．

以上の2つの理由から，一貫性制約の保証範囲を限定するのに伴い設計情報に矛盾が生じても，後でチェックを行い，それに基づきツールで一貫性を回復可能であれば良いという考え方を採っている．

本論文では，WS の集合（通常ビュー）をツールの参照の範囲かつユーザ権限の範囲に設定していることから，一貫性制約の影響範囲を WS の集合に限定可能である．これらのツールを使用する際には，WS の使用宣言を行った後，編集操作を行うことから，一貫性制約は，使用宣言がなされている WS の範囲で保証すれば良い．

ここで，データ管理者が統一コードを削除する上記の例をさらに詳しく見ると，上記の例は，使用宣言の有無に基づきさらに3つの場合に分けられる．①統一コードを使用している WS「DB」を使用宣言せずに，WS「統一コード」のみを使用宣言し，統一コードの削除を試みると，参照一貫性の波及が，「DB」に及ばず，参照一貫性が崩れる．②①において，「DB」を参照モードで使用宣言し同様の削除を行う．DB 情報の更新はやはりできず，参照一貫性が崩れる．③①において，「DB」を更新モードで使用宣言して削除を試みる．この時のみ削除の影響が「DB」に及び，WS にまたがる一貫性が保証される．①，②の状況を避ける必要がある場合には，それを禁じるように関連する WS を指定したビューを使用する必要がある．

一方，①，②の状況を許した時は，ディクショナリに矛盾が起こった時どう解決するのかが問題である．これについては，両方の WS をアクセス可能なツールが，統一コードの主キーと，DB 情報に含まれた参照キーを比較することで対処できる．また，IRDS でこのようなチェック機能を提供することも可能である．

4. ディクショナリ設計時の考慮点

本論文の場合，ディクショナリの設計にはリレーショナルデータベースの設計手法を適用できる．しかし，一貫性制約の波及を保証する範囲を WS としたことに伴い，次の制約が必要になる．

①まず，WS 内では少なくとも一貫性を保証する必要があるため，他の WS の使用宣言の有無にかかわらず WS 内に矛盾が生じないようなディクショナリ構成とする必要がある．例えば，図6に示す参照一貫性において，テーブル「DB」の削除に伴い，異なる WS に属するテーブル「テーブル」に削除の波及が起こり，さらにそれが，「DB」と同一の WS のテーブル「データ項目」の削除に影響する設計を行うと，「テーブル」を含む WS が使用宣言されていない場合，「DB」，「データ項目」を含む WS に矛盾が起こる可能性がある．したがって，一貫性制約の影響の連鎖が，一度 WS を出て，再び同一の WS に戻るディクショナリ設計を行ってはいけないという制約を課す．

②また，WS をまたがる一貫性制約において矛盾が生じる場合には，それがチェック可能なディクショナリを設計する必要がある．参照一貫性については，3.3 節に示したように主キーと参照キーの相互チェックによりお互いのタプルの過不足をチェック可能である．しかし，主キーが更新されれば，その値からどの値に更新されたかをチェックができない．そこで，IRDS 全体で一意な番号を付与しその番号でも対応を取れるようにする．これにより，主キーが異なっても，番号が同じなら変更されたことがわかる．

この番号払い出しの機能を IRDS で提供する．この番号はホスト，ローカルシステム全体で一意となるよう払い出す．一方，異なるシステムから設計情報を移行する場合には，IRDS 内の払い出し番号と矛盾しないように移行ツールで新しい番号に付け替える必要がある．

5. IRDS の実現方法

IRDS の実現は，自前での構築も可能であるが，ISO の IRDS が SQL を基本としていることから，リ

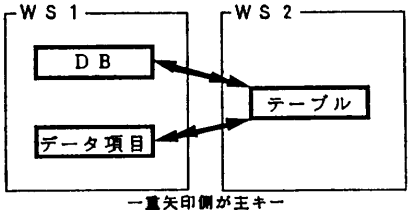
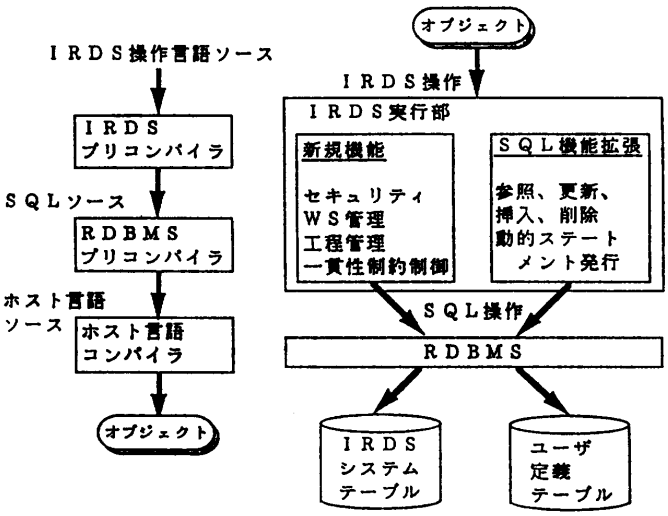


図 6 一貫性が保てないディクショナリ定義の例
Fig. 6 Inconsistent dictionary definition.

レーショナルDB上への構築が容易であると考えた．図7にIRDSの実現例を示す．

IRDSに固有で，新規の機能は，セキュリティ機能，WS管理機能，工程管理機能である．また，既にSQLに存在する，テーブルの参照更新機能，および，動的ステートメント発行機能は，使用宣言されているWSに対して操作を行っているかのチェック機能の追加，および，指定したWSに操作範囲を限定する機能変更が必要なため，機能拡張する．このコマンドをホスト言語に埋め込みIRDSでプリコンパイルする方式で実現する場合には，前者の新規機能は，その機能を実現するサブルーチンへの展開により，後者の機能は，チェックを行うサブルーチンとSQLを含むホスト言語への展開により実現可能である．



(a) プリコンパイル時 (b) 実行時
(a) Precompilation phase. (b) Execution phase.

図7 IRDSの実現例
Fig. 7 An implementation of IRDS.

6. おわりに

本論文では，DB設計支援分野からの要求に基づきIRDSの必要機能を見直し，新たな機能提案を行った．プロジェクト管理のためのグループ化の概念としては，複合オブジェクトが十分機能しないことを示し，代わりに，ISOのIRDSで提案されているワーキングセットを基本に操作機能の具体化，および，確認を行った．次に，従来の工程管理機能のみでは設計情報の品質まで管理できないことを示し，工程管理機能とは独立な品質管理機能を実現した．次に，IRDSでは，一貫性制約の保証範囲をワーキングセットに限定すべきことを示した．また，DBに対する設計手法に加えてディクショナリで考慮すべき設計時の工夫点を示した．最後に，リレーショナルDBMSを用いたIRDSの実現法を示した．

これらの機能を具体化し，データ標準化ツール，DB論理物理設計ツールで動作確認を行うことにより，提案機能が有効であることを確認した．これらのツールとIRDSの一部機能は，既にSUN上でリレーショナルデータベース管理システムであるINFORMIXを用いて開発済みであり，運用に供している．DB設計支援システムの全体規模は現在C言語で140Kステップ，そのうちIRDSは10Kステップである．現在，データ標準化ツールで管理しているデータ項目情報が13,000件，標準用語情報が，2,500件存在する．

本論文での検討の結果に基づくIRDSの今後の検討課題については，次の3点が挙げられる．最初の検討課題は，IRDSによるDB設計支援システム実現において，①どのように工程を設定すべきか，および，②工程，品質表示名の追加，変更に影響を受けにくいツールの設計方法はどうあるべきか，といったIRDSの使用方法に関するものである．これらについては，今後もツールの構築を通じて検討を重ねたい．

2番目の検討課題は，WSの機能拡充に関するものである．今回提案したWS操作機能は，DB設計支援のための必要最小限の機能となっている．一方，ISOのIRDS国内検討サブグループでは，WS間の包含関係，および，あるWSの情報があたかも他のWSの情報であるかのように参照，更新できるWS間の参照関係の定義，操作機能について議論されており，そのような機能の必要性と具体化を今後検討する必要がある．

最後の検討課題は，IRDSにおける複合オブジェクトの実現に関するものである．本論文ではWSと複合オブジェクトは別の物と捉え，複合オブジェクトについては議論しなかった．しかし，設計情報の構造の表現のためには複合オブジェクトは不可欠であり，これを今後どのようにIRDSに取り込むかの検討が必要である．この対処策には，現在のリレーショナルモデルを拡充するという考え方もあるが，既にオブジェクト指向モデルでは複合オブジェクトが提供されていることから，オブジェクト指向モデルの標準化を待っ

て，IRDS のモデルをリレーショナルモデルからオブジェクト指向モデルに移行してゆくことも考えている．この場合でも，WS の概念は目的が異なるので残すべきだと考えている．

最後に，ここで得られた知見は ISO の IRDS 国内検討サブグループでの検討の一助としたい．

謝辞 ISO の IRDS 標準化のため日頃検討を共にし，示唆を与えていただく SC 21/WG 3/RMDM+IRDS 国内サブグループの皆様に感謝します．また，本研究の機会と適切なアドバイスを与えていただいた NTT 情報通信網研究所の田中豪主幹研究員に感謝します．

参考文献

1) American National Standard for Information Systems—Information Resource Dictionary System-Services Interface, ISO/IEC JTC1/SC21 WG 3 N 955 (1989).
2) ATIS—A Tools Integration Standard, US National Body, ISO/IEC JTC 1/SC 21 WG 3 N 1020 (1990).
3) Braegger, R. P., Dudler, A. M., Rebsamen, J. and Zehnder, C. A.: GAMBIT; An Integrated DB Design Tool for Data Structures, Integrity Constraints, and Transactions, *IEEE Trans. Softw. Eng.*, Vol. SE-11, No. 7, pp. 574-583 (1985).
4) Carey, M. J., DeWitt, D. J. and Vandenberg, S. L.: A Data Model and Query Language for EXODUS, *Proc. ACM SIGMOD*, pp. 413-423 (1988).
5) Chou, H. T. and Kim, W.: A Unifying Framework for Version Control in a CAD Environment, *Proc. 12th VLDB Conf.*, pp. 336-344 (1986).
6) Chroust, G., Goldman H. and Gschwandtner, O.: The Role of Work Management in Application Development, *IBM Syst. J.*, Vol. 29, No. 2, pp. 189-208 (1990).
7) Dittrich, K. R., Kotz, A. M. and Mulle, J. A.: An Event/trigger Mechanism to Enforce Complex Consistency Constraints in Design Databases, *ACM SIGMOD Record*, Vol. 15, No. 3, pp. 22-36 (1986).
8) Durell, W. R.: データ資源管理，日経マグロウヒル (1987).
9) Fong, E. N. and Goldfine, A. H.: Data Base Directions; Information Resource Management —Making it Work, *ACM SIGMOD Record*, Vol. 15, No. 3, pp. 3-10 (1986).
10) Gardarin, G., Cheiney, J. P. and Kiernan, J.: Extending a Relational DBMS to Support Complex Objects, *IEEE 2nd Int. Conf. Data Knowl. Syst. Manuf. Eng.*, pp. 131-137 (1989).
11) Goldfine, A. and Konig, P.: A Technical Overview of the Information Resource Dictionary System, NBSIR 88-3700, National Bureau of Standard (1988).
12) Haynie, M. N.: A DBMS for Large Design Automation Databases, *Proc. ACM SIGMOD*, pp. 269-276 (1988).
13) Hudson, S. E. and King, R.: Object-Oriented Database Support for Software Environments, *Proc. ACM SIGMOD*, pp. 491-503 (1987).
14) Hong, S. and Maryanski, F.: Using a Meta Model to Represent Object-Oriented Data Models, *IEEE Int. Conf. Data Engineering*, pp. 11-19 (1990).
15) Information Resource Dictionary System (IRDS) Services Interface, Working Draft, Rev. 11, ISO/IEC JTC1/SC21/N4895 (1990).
16) Information Technology-Information Resource Dictionary System (IRDS) Framework, ISO/IEC 10027 (1990).
17) Katz, R. H. and Chang, E.: Managing Change in a Computer-Aided Design Database, *Proc. 13th VLDB Conf.*, pp. 455-462 (1987).
18) Kim, W., Bertino, E. and Garza, J. F.: Composite Objects Revisited, *Proc. ACM SIGMOD*, pp. 337-347 (1989).
19) Kim, W., Chou, H.T. and Banerjee, J.: Operations and Implementation of Complex Objects, *Int. Conf. Data Engineering*, pp. 626-632 (1987).
20) Kim, W., and Chou, H. T.: Versions of Schema for Object-Oriented Databases, *Proc. 14th VLDB Conf.*, pp. 148-159 (1988).
21) Kim, W., Lorie, R., McNabb, D. and Plouffe, W.: A Transaction Mechanism for Engineering Design Databases, *10th VLDB Conf.*, pp. 355-362 (1984).
22) Kitagawa, H. and Ohbo, N.: Design Data Modeling with Versioned Conceptual Configuration, *IEEE 13th Annual Int. Computer Software and Application Conference*, pp. 225-233 (1989).
23) Lafue, G. M. and Smith, R. G.: Implementation of a Semantic Integrity Manager with a Knowledge Representation System, *Expert Database Systems*, pp. 333-350, The Benjamin/Cummings Publishing Co. Inc. (1986).
24) Mark, L. and Roussopoulos, N.: Metadata Management, *IEEE Comput.*, Vol. 19, No. 12, pp. 26-36 (1986).
25) Mitschang, B.: Extending the Relational Algebra to Capture Complex Objects, *Proc. 15th VLDB Conf.*, pp. 297-305 (1989).

26) Morgenstein, M.: The Role of Constraints in Databases, Expert Systems, and Knowledge Representation, *Expert Database Systems*, pp. 351-368, The Benjamin/Cummings Publishing Co. Inc. (1986).
27) Sagawa, J. M.: Repository Manager Technology, *IBM Syst. J.*, Vol. 29, No. 2, pp. 209-226 (1990).
28) Shepherd, A. and Kerschberg, L.: Constraint Management in Expert Database Systems, *Expert Database Systems*, pp. 309-331, The Benjamin/Cummings Publishing Co. Inc. (1986).
29) 関根 純, 川下 満, 鈴木健司：ネーミング手法と支援ツール, 電子情報通信学会データ工学研究会, DE89-4 (1989).
30) 先進ユーザがつかんだデータ項目名標準化の糸口, 日経コンピュータ, pp. 105-113 (1987. 9. 14).
31) 三菱銀行のデータディクショナリによる集中管理, 事務管理, Vol. 26, No. 9, pp. 97-104 (1987).

(平成3年5月 1 日受付)
(平成4年1月17日採録)

関根 純（正会員）

1958 年生. 1980 年東京大学工学部計数工学科卒業. 1982年同大学院修士課程修了. 同年, 日本電信電話公社横須賀通信研究所に入社. データベース管理システム, マルチメディアデータベースの研究実用化に従事. 現在, データベース設計支援, 情報資源管理の研究実用化を行う. 情報規格調査会 SC 21/WG 3/RMDM+IRDS サブグループ委員. ACM 会員.

川下 満（正会員）

1951 年生. 1975 年山口大学大学院工学研究科修士課程修了. 同年日本電信電話公社入社. データベースの分散処理方式, マルチメディア情報検索等の研究実用化に従事. 現在 NTT 情報通信網研究所にて, データベースの設計法, および, その支援システムの研究実用化を行っている.

中川 優（正会員）

昭和 22年生. 昭和 45年大阪大学基礎工学部制御工学科卒業. 昭和 47年同大学院修士課程修了. 同年, 日本電信電話公社武蔵野通研入所. OS, DBMS の実用化, および, 自然言語理解, 知識処理の研究に従事. 現在, NTT 情報通信網研究所データベース研究部にて, データベース設計法, 情報資源管理の研究実用化に従事. 主幹研究員. 工学博士. 人工知能学会, 電子情報通信学会各会員.